SYSTEM AND METHOD FOR PRODUCING A DETECTOR POSITION MAP

BACKGROUND OF THE INVENTION

[0001] The present invention relates generally to an imaging device, and more particularly to a system and method for mapping a detected radiation event to a particular detector in a medical imaging device such as a positron emission tomography scanner.

[0002] A positron emission tomography (PET) scanner detects gamma rays which emanate from the patient. In a PET scan, the patient is initially injected with a radiopharmaceutical, which is a radioactive substance such as FDG ([^{18}F] fluorodeoxyglucose) which emits positrons as it decays. Once injected, the radiopharmaceutical becomes involved in certain known bodily processes such as glucose metabolism or protein synthesis, for example. The emitted positrons travel a very short distance before they encounter an electron, at which point an annihilation event occurs whereby the electron and positron are annihilated and converted into two gamma rays. Each of the gamma ray has an energy of 511 keV, and the two gamma rays are directed in nearly opposite directions. The two gamma rays are detected essentially simultaneously by two of the detector crystals (also commonly referred to as “scintillators” or “scintillator crystals”) in the PET scanner, which are arranged in rings around the patient bore. The simultaneous detection of the two gamma rays by the two detector crystals is known as a “coincidence event.” The millions of coincidence events which are detected and recorded during a PET scan are used to determine where the annihilation events occurred and to thereby reconstruct an image of the patient.

[0003] One of the challenges in designing a high resolution PET scanner relates to the space requirements of the electronics associated with the detector crystals, in particular the photomultiplier tubes (PMTs) which are situated behind the detector crystals. The function of the photomultiplier tubes is to receive photons produced by the scintillator crystals and to generate an analog signal with a magnitude representative of the number of photons received. The photomultiplier tubes typically cannot be diminished in size beyond a certain point, so that generally each photomultiplier tube is situated behind a number of smaller detector crystals. For example, a detector module in a PET scanner may comprise a 2 x 2 array of photomultiplier tubes situated behind a 6 x 6 array of scintillator crystals. The photons generated by a detector crystal generally spread out to a certain extent and travel into adjacent detector crystals such that each of the four photomultiplier tubes typically receives a certain number of photons as a result of a gamma ray hitting a single detector crystal.

[0004] In response to a scintillation event, each PMT produces an analog signal which rises sharply when a scintillation event occurs then tails off exponentially. The relative magnitudes of these analog signals are used to calculate a coordinate pair (x, z) which indicates the position at which the gamma ray was incident on the crystal detector array. The (x, z) coordinate pair is mapped to a specific detector crystal in the detector array using a look-up table. The data in the look-up table may be referred to as a detector position map. For each pair of coordinates (x, z), the detector position map maps that pair of coordinates to one of the detectors in the array of detectors.

[0005] With ideal hardware, the detector position map would simply be a rectangular grid corresponding to the geometry of the detector crystals, where each detector crystal would have an equal area on the grid. However, this simple mapping scheme is generally not accurate due to non-linearity in the analog parts of the detector and manufacturing differences between detector crystals. For example, if a simple rectangular grid were used as the detector position map, a certain percentage of (x, z) coordinate pairs would be mapped to the wrong detector crystal.

[0006] The present invention provides a system and method for producing a detector position map which accurately maps the detected coordinates (x, z) to the correct detector crystal.

SUMMARY

[0007] According to one embodiment, the invention relates to a method of generating a detector position map for an array of detectors. The detector position map comprises a map which maps measured coordinates from a detection event to the detector in the array which detected the detection event. The method comprises the steps of illuminating an array of detectors with a source of radiation to generate a histogram, the histogram comprising an event count as a function of two dimensions, the two dimensions corresponding to a face of the array of detectors, wherein the histogram comprises a plurality of first peaks; modifying the histogram to comprise a plurality of second peaks, wherein the second peaks have a greater degree of isolation from each other than the first peaks; and for each detector, determining a region on the detector position map which corresponds to the detector, each region being based on a position of one of the second peaks.

[0008] According to another embodiment, the invention relates to a system for generating a detector position map for an array of detectors. The detector position map comprises a map which maps measured coordinates from a detection event to the detector in the array which detected the detection event. The system comprises a memory; and a processor which receives a plurality of event data packets, each event data packet comprising a coordinate pair; stores the plurality of event data packets in the form of a histogram in the memory, the histogram comprising an event count as a function of two dimensions, the two dimensions corresponding to a face of the array of detectors, wherein the histogram comprises a plurality of first peaks; modifies the histogram to comprise a plurality of second peaks, wherein the second peaks have a greater degree of isolation from each other than the first peaks; for each detector, determines a region on the detector position map which corresponds to the detector, each region being based on a position of one of the second peaks; and stores each region in the memory to create the detector position map.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Figure 1 is a drawing of an imaging system according an exemplary embodiment of the invention;

[0010] Figure 2 is a schematic diagram of the imaging system of Figure 1;

[0011] Figure 3 is a drawing of a detector module which forms part of the imaging system of Figure 1;

[0012] Figure 4 is a 3D image of a compressed position histogram according to an exemplary embodiment of the invention;

[0013] Figure 5 is a top view of the compressed position histogram of Figure 4;

[0014] Figure 6 is a flow chart showing steps in an example of a method for generating a detector position map;

[0015] Figure 7 is an example of a 3 x 3 low pass filter kernel;

[0016] Figure 8 depicts two examples of 3 x 3 Laplacian filter kernels;

[0017] Figure 9 is an example of a 9 x 9 Laplacian filter kernel;

[0018] Figure 10 is a 3D image of a histogram which has been convolved with a Laplacian filter;

[0019] Figure 11 is an image of the histogram of Figure 10 after a threshold operation has been performed;

[0020] Figure 12 is a top view of the histogram of Figure 11;

[0021] Figure 13 is a map of the peak apex pixel locations of the peaks of Figure 12;

[0022] Figure 14 is an example of a numbering scheme for a 6 x 6 detector array; and

[0023] Figure 15 is a detector position map generated from the peak apex pixel locations shown in Figure 13.

DETAILED DESCRIPTION

[0024] The operation of a PET scanner will be described, followed by a description of the generation of the detector position map used by the scanner to map events to a particular detector in the array.

[0025] Figure 1 illustrates an example of a PET scanner 1 which includes a gantry 10 supporting a detector ring assembly 11 about a central opening or bore 12. The detector ring assembly 11 is circular in shape and is made up of multiple detector rings (not shown) that are spaced along a central axis 2 to form a cylindrical detector ring assembly. According to one embodiment, the detector ring assembly 11 includes 24 detector rings spaced along the central axis 2. A patient table 13 is positioned in front of the gantry 10 and is aligned with the central axis 2 of the detector ring assembly 11. A patient table controller (not shown) moves the table bed 14 into the bore 12 in response to commands received from an operator work station 15 through a serial communications link 16. A gantry controller 17 is mounted within the gantry 10 and is responsive to commands received from the operator work station 15 through a second serial communication link 18 to operate the gantry.

[0026] As shown in Figure 2, the operator work station 15 includes a central processing unit (CPU) 50, a display 51 and a keyboard 52. Through the keyboard 52 and associated control panel switches, the operator can control the calibration of the PET scanner, its configuration, and the positioning of the patient table for a scan. Similarly, the operator can control the display of the resulting image on the display 51 and perform image enhancement functions using programs executed by the work station CPU 50.

[0027] Each detector ring of the detector ring assembly 11 is comprised of a number of detector modules 20. According to one embodiment, each detector ring includes 72 detector modules 20. Each detector module 20 comprises a number of individual detector crystals. As shown in Figure 3, each detector module 20 may comprise a 6 x 6 matrix of 36 detector crystals 21. Figure 3 shows an arrangement of a 6 x 6 array of detector crystals 21 which are situated in front of four photomultiplier tubes (PMTs) 22.

[0028] During a PET scan, a gamma ray is incident upon one of the detector crystals 21. The detector crystal 21, which may be formed of bismuth germanate (BGO), for example, converts the gamma ray into a number of photons which are received and detected by the photomultiplier tubes. The photons generated by a detector crystal 21 generally spread out to a certain extent and travel into adjacent detector crystals such that each of the four photomultiplier tubes 22 receives a certain number of photons as a result of a gamma ray hitting a single detector crystal 21.

[0029] In response to a scintillation event, each PMT 22 produces an analog signal 23A-23D on one of the lines A-D shown in Figure 3 which rises sharply when a scintillation event occurs then tails off exponentially. The relative magnitudes of the analog signals are determined by the position in the 6 x 6 detector crystal matrix at which the scintillation event took place. The energy of the gamma ray which caused the scintillation event determines the total magnitude of these four signals.

[0030] As shown in Figure 2, a set of acquisition circuits 25 is mounted within the gantry 10 to receive the four signals from each of the detector modules 20 in the detector ring. The acquisition circuits 25 determine the event coordinates within the block of detector crystals 21 by comparing the relative signal strengths as follows:

$$x = (A+C)/(A+B+C+D)$$

$$z = (A+B)/(A+B+C+D)$$

[0031] These coordinates (x, z), along with the sum of all four signals (A+B+C+D) are then digitized and sent through a cable 26 to an event locater circuit 27 which may be housed in a separate cabinet 28. Each acquisition circuit 25 also produces an event detection pulse (EDP) which indicates the exact moment the scintillation event took place. Of course, the above-described configuration of detector crystals and modules is merely an example. Other configurations, sizes, and numbers of detector crystals and detector modules can be used, as will be appreciated by those skilled in the art.

[0032] The event locator circuits 27 form part of a data acquisition processor 30 which periodically samples the signals produced by the acquisition circuits 25. The data acquisition processor 30 has an acquisition CPU 29 which controls communications on the local area network 18 and a bus 31. The event locator circuits 27 assemble the information regarding each valid event into a set of digital numbers that indicate precisely when the event took place and the identity of the detector crystal 21 which detected the event. The event locator circuits 27 use a detector position map to map the coordinate pair (x, z) to the detector crystal which detected the event. The detector position map is a two dimensional look-up table that maps a singles event (x, z) coordinate position to a detector ID number. An example of a detector position map is shown in Figure 15.

[0033] The event data packets are transmitted to a coincidence detector 32 which is also part of the data acquisition processor 30. The coincidence detector 32 accepts the event data packets from the event locator circuits 27 and determines if any two of them are in coincidence. Coincidence is determined by a number of factors. First, the time markers in each event data packet must be within a specified time period of each other, e.g., 12.5 nanoseconds, and second, the locations indicated by the two event data packets must lie on a straight line which passes through the field of view (FOV) in the scanner bore 12. Events which cannot be paired are discarded, but coincident event pairs are located and recorded as a coincidence data packet that is transmitted through a serial link 33 to a sorter 34. The format of the coincidence data packet may be, for example, a thirty-two bit data stream which includes, among other things, a pair of digital numbers that precisely identify the locations of the two detector crystals 21 that detected the event. For a detailed description of an example of a coincidence detector 32, reference is made to U.S. Pat. No. 5,241,181 entitled "Coincidence Detector For A PET Scanner."

[0034] The sorter 34, which may comprise a CPU and which forms part of an image reconstruction processor 40, receives the coincidence data packets from the coincidence detector 32. The function of the sorter 34 is to receive the coincidence data packets and generate from them memory addresses for the efficient storage of the coincidence data. The set of all projection rays that point in the same direction (θ) and pass through the scanner's field of view is a complete projection, or "view." The distance (R) between a particular projection ray and the center of the field of view locates that projection ray within the view. As shown in Figure 2, for example, an event 50' occurs along a projection ray 51' which is located in a view at the projection angle θ and the distance R. The sorter 34 counts all of the events that occur on this projection ray (R, θ) during the scan by sorting out the coincidence data packets that indicate an event at the two detector crystals 21 lying on this projection ray. During an emission scan, the coincidence counts are organized in memory 43, for example as a set of two-dimensional arrays, one for each axial image, and each having as one of its dimensions the projection angle θ and the other dimension the distance R. This θ by R map of the measured events may be referred to as a sinogram array 48. The sorter 34 may also organize the coincidence events into other data formats, such as the projection plane format described in U.S. Patent No. 6,462,342. In the projection plane format, additional variables are used to define coincidence events which are detected by pairs of detector crystals in non-adjacent detector rings.

[0035] Coincidence events occur at random and the sorter 34 determines the θ and R values from the two crystal addresses in each coincidence data packet and increments the count of the corresponding sinogram array element. At the completion of the emission scan, the sinogram array 48 stores the total number of annihilation events which occurred along each ray. The array processor 45 reconstructs an image from the data in the sinogram array 48. First, however, a number of corrections are made to the acquired data to correct for measurement errors such as those caused by attenuation of the gamma rays by the patient, detector gain nonuniformities, random coincidences, and integrator deadtime. Each row of the corrected sinogram array is then Fourier transformed by the array processor 45 and multiplied by a one-dimensional filter array. The filtered data is then inverse Fourier transformed, and each array element is backprojected to form the image array 46. The image CPU 42 may either store the image array data on disk or tape (not shown) or output it to the operator work station 15.

[0036] An example of a method for generating a detector position map which is used to map the coordinate pairs (x, z) to the detector crystal which detected the event will now be described. The method can be carried out with the system shown in

Figure 2. For example, the acquisition CPU 29 can be used to perform the processing steps in the exemplary method. The position histogram can be created with the event locator circuits 27, and the resulting detector position map can be loaded into the event locator circuits 27. Figure 6 is a flow chart showing the steps for generating a detector position map using a position histogram as input data to produce the detector position map as output data according to an exemplary embodiment of the invention. The steps in the method will be explained in detail in the following discussion.

[0037] Initially, the acquisition hardware of the PET scanner is configured to create a position histogram. The position histogram is created by illuminating the detector array with a spatially uniform gamma ray flux. The position histogram is a histogram of the (x, z) coordinate positions of each singles event which occurs during the illumination with the uniform gamma ray flux. The position histogram is ultimately converted into a detector position map.

[0038] The form of a typical position histogram viewed as a 3D image is shown in Figure 4, with the vertical axis indicating the number of singles events that mapped to each (x, z) location during the illumination with the uniform gamma ray flux. The histogram shown in Figure 4 is a compressed version of the raw histogram data. In general, each detector crystal produces a footprint in the position histogram that appears as a mountain peak. As shown in Figure 4, there can be a wide variance in both peak height and the valley heights between the peaks. The detector that produced Figure 4 comprised a 6 x 6 matrix of crystals, giving 36 major peaks. The methods described herein, however, are not limited to this geometry.

[0039] After the position histogram data have been generated, optionally the data can be compressed, if desired, to reduce processing time. This step may be advantageous if the nominal size of the position histogram is relatively large. As an example, if the size of the position histogram is 256 x 256, it can be compressed to a 128 x 128 compressed histogram by summing four adjacent pixels in the 256 x 256 histogram to produce one pixel in the 128 x 128 compressed histogram. The compression operation will also have a smoothing effect on the data. Figure 4 is a 128 x 128 histogram which was derived by compressing a 256 x 256 position histogram. Figure 5 is the same image, but viewed from the top looking down. The following discussion assumes that the nominal histogram size is 256 x 256 and that it has been compressed to 128 x 128 for processing.

[0040] A typical position histogram may contain a significant amount of noise that may advantageously be reduced with a low pass filter. For example, the compressed position histogram shown in Figure 4 may be smoothed to reduce noise by a convolution with a low pass filter. An example of a suitable low pass filter is shown in Figure 7. As is well known in the art, convolution is carried out by placing the filter on the histogram, multiplying each value in the low pass filter with the corresponding (overlapping) pixel value in the histogram, adding the products together to get a sum, and replacing the cell of the histogram corresponding to the center pixel of the low pass filter with the sum of the products. The low pass filter is then moved to the next pixel on the histogram and the process is repeated. After the low pass filter has been applied to each pixel in the compressed position histogram, the resulting data, which may be referred to as a smoothed histogram, is smoothed with respect to the compressed position histogram.

[0041] In the next step of the exemplary method, the data of the smoothed histogram is convolved with a Laplacian filter. The Laplacian filter can be used to separate the histogram peaks from the valleys. A Laplacian filter calculates the second derivative, which is the curvature of the image over the area of the filter kernel. Examples of 3 x 3 Laplacian filters are illustrated in Figure 8. An example of a 9 x 9 Laplacian filter is shown in Figure 9. In the 9 x 9 Laplacian filter of Figure 9, the empty cells have a value of zero. Convolution of the smoothed histogram with the Laplacian filter is carried out in the same way as described above with respect to the low pass filter.

[0042] Figure 10 shows the result of a Laplacian convolution of the smoothed histogram from the previous step. From Figure 10 it can be seen that the Laplacian filter transforms the histogram valleys into negative peaks, while retaining the general form and location of the histogram peaks. This operation results in a more defined separation between the histogram peaks that are close together.

[0043] The next step in the exemplary method comprises a threshold determination. This step can remove noise from the Laplacian filter output. In this step, each pixel in the Laplacian histogram is examined to determine whether it is less than a specified threshold value. If the pixel value is less than the threshold value, then the pixel value is set to zero. The threshold value, according to one embodiment, is a positive number on the order of 100. Figure 11 illustrates the output from this step using a threshold value of 125. The output histogram may be referred to as a thresholded histogram. Figure 12 is a top view of Figure 11. A comparison of Figure 12 with the image in Figure 5 illustrates the separation of peaks and the reduction of noise in the thresholded histogram.

[0044] According to one embodiment, the threshold value is selected based on the total number of counts in the uncompressed position histogram using the empirical formula:

$$\text{Threshold} = 0.00013158 * \text{TotalCount}$$

Where TotalCount is sum of each pixel value in the position histogram. A large threshold value will give better isolation of peaks and reduce the amount of data for the remaining steps. However, since there may be a wide variance in peak height, too large a threshold value can eliminate valid peaks.

[0045] After the thresholded histogram has been generated, the next step of the exemplary method involves producing a list of peak apex candidate locations. The term "peak apex" refers to the highest point on a peak. This step involves examining each pixel in the thresholded histogram to determine whether it is a peak apex. The following pseudo code illustrates the process:

```
For each pixel in the thresholded histogram, do the following:
        if the pixel is zero,
                skip the pixel, since its not contained within a peak.
        else
                Look at the eight surrounding pixels. If the value of the pixel is
                greater than its eight neighbors,
                Then save it as a peak apex candidate.
```

The output of this step is a list of peak apex candidate locations.

[0046] Preferably, the number of peak apex candidates from the thresholding step would equal the number of detector crystals in the detector array. If the number of peak apex candidates is the same as the number of detector crystals, the next step, eliminating false peaks, is not necessary. If there are fewer peak apex candidates than detector crystals in the array, then the threshold value may be too large. The threshold value can be reduced such that the thresholded histogram contains at least as many peak apex candidates as the number of detectors in the array.

[0047] If there are more peak apex candidates than detector crystals, then false peak apex candidates can be eliminated by searching the list of peak apex candidates for the two closest pairs and eliminating the candidate with the smaller value. This process assumes that two peak apex candidates that are close together are part of the same peak. The process can be continued until the number of peak apex candidates equals the number of detectors in the array. The following pseudo code describes the method:

```
Do the following until the number of candidates equals the number of crystals:
    Search the list of peak apex candidates for the two closest pairs using
    Euclidean distance as the metric.
    Eliminate the candidate with the smallest pixel value.
```

Figure 13 is a resulting image of the peak apex pixels that correspond to the position histogram in Figure 4.

[0048] Another method can be implemented to determine the peak apex pixels, according to an exemplary embodiment of the invention. This feature involves converting the peak apex candidates to decimal numbers. After the peak apex candidates have been converted to decimals, a centroid is calculated for a desired peak neighborhood. The neighborhood can be defined, for example, as the region within a certain distance of the peak apex pixel, e.g., all pixels within a 10-pixel radius of the peak apex pixel. The centroid is analogous to a center of mass, assuming that the height of the pixels represents a mass. By calculating the centroid for a desired peak neighborhood, false peaks that are close to the centroid peak can be eliminated.

[0049] As is evident from Figures 11 and 12, the original position histogram has been modified in such a way as to enhance the isolation of its peaks. Figure 12 shows, for example, that the peaks have been isolated. The Laplacian can provide the advantage of accentuating the separation between peaks by, for example, forcing the valleys between the peaks to be zero, after thresholding. The average cross section of the peaks of the original position histogram has also been reduced. This modification of the peaks facilitates the subsequent determination of the peak apex pixels.

[0050] After the peak apex pixels have been identified, the next step involves assigning each peak apex to a detector crystal. Figure 14 depicts an example of a detector crystal numbering scheme. Each peak apex pixel identified in the previous step is assigned one of the detector crystal identification numbers (1-36) shown in

Figure 14. The result of this step is a 2D array, where each cell in the array corresponds to a cell in Figure 14 and contains the (x, z) coordinates for the peak apex pixel and the ID number (e.g., 1-36) of the corresponding detector crystal. This array may be referred to, for example, as the CrystalPeaks array.

[0051] Initially, the (x, z) coordinates for each peak apex pixel are stored, for example, in a 1D array. Each entry in this 1D array includes an x coordinate and a z coordinate. This array may be referred to, for example, as the PeakApex array.

[0052] The detector ID numbers shown in Figure 14 are then stored in the 2D CrystalPeaks array. For example, CrystalPeaks [0][0].Id is set to 1, CrystalPeaks[3][3].Id is set to 22, CrystalPeaks[5][5].Id is set to 36, etc. As will be appreciated by those skilled in the art, any other desired numbering format, e.g., hexadecimal, may be used for the detector ID numbers.

[0053] Next, each (x, z) coordinate pair contained in the 1D PeakApex array is randomly stored in a cell of the 2D CrystalPeaks array.

[0054] The next step in the process comprises a sorting routine in two dimensions, x and z. A conventional bubble sort routine can be used. This method makes use of the realization that the peak apex locations (x, z) can be sorted based on location of both their x and z positions. For example, a peak apex assigned to crystal 1 will have a lesser x value than a peak apex assigned to crystal 2. Likewise, a peak apex assigned to crystal 1 have a lesser z value than a peak apex assigned to crystal 7. This sorting operation may be applied to sort the x data in the array into increasing order of the x coordinate. This sorting operation may then be applied to sort the z data in the array in increasing order of the z coordinate. In the resulting sorted array, the coordinate pairs (x, z) representing the peak apex pixels are assigned to the crystal ID numbers in the configuration shown in Figure 14.

[0055] The following pseudo code illustrates an exemplary method of assigning each peak apex to a detector crystal. The pseudo code uses the following definitions:

"CrystalsInX" is the number of detector crystals in the x direction. In this example it is 6.

"CrystalsInZ" is the number of detector crystals in the Z direction. In this example it is 6.

"CrystalPeaks[x][z]" is a 2D array that will contain the location coordinates (x, z) and crystal ID number (e.g., 1-36) for crystal peak. The size of the array is CrystalsInX by CrystalsInZ. Each entry of the array has a location given by CrystalPeaks[x][z].X and by CrystalPeaks[x][z].Z and a crystal ID number given by CrystalPeaks[x][z].Id.

"PeakApex[i]" is a list of peak apex locations created in the previous step. Each entry has a location given by PeakApex[i].X and PeakApex[i].Z.

The pseudo code is as follows:

```
First, mark all CrystalPeaks[x][z].Id entries to reflect the numbering in Figure
14. For example, CrystalPeaks[0][0].Id = 1, CrystalPeaks[5][5].Id =36, etc.
Next, randomly assign a peak location from the PeakApex[I] array to a
location in the CrystalPeaks[x][z] array.
notDone = TRUE;
While (notDone == TRUE)
begin
    notDone = FALSE;
    Sort the x data in the CrystaPeaks[x][z] array in increasing order. If
    anything was moved, set notDone = TRUE.
    Sort the z data in the CrystaPeaks[x][z] array in increasing order. If
    anything was moved, set notDone = TRUE.
end;
```

[0056] After each peak apex pixel has been associated with a detector crystal and assigned a crystal ID number, the remainder of the pixels in the detector position map are assigned a crystal ID number, based on their proximity to a peak apex pixel. Each pixel in the detector position map is assigned the crystal ID number of the closest peak apex, using Euclidean distance as the metric.

[0057] For each pixel in the detector position map, the process of determining which peak apex is closest involves calculating a distance indication between the pixel in question and each peak apex pixel. The distance calculation is carried out by calculating the sum of the squares of the change in x and the change in z between the pixel in question and each peak apex pixel. To calculate the distance, the square root of the sum of the squares of the change in x and the change in y can be calculated. However, the square root step can be omitted, if desired, because the objective is simply to find the smallest magnitude.

[0058] An efficiency check can be included, if desired, to skip the step of calculating distances if the change in x and the change in z are greater than a predetermined distance relative to the detection position map, e.g., greater than 1/3 the total detector position map dimension. In such case, the distance calculation can be skipped because it may be safely concluded that the peak apex pixel in question is not the closest one to the pixel in question.

[0059] The following pseudo code provides an example of the process for associating a peak apex pixel and crystal ID number to each pixel on the detector position map. The following definitions can be used for the pseudo code.

"Pixel" is the current detector position map pixel under consideration. It has a location given by pixel.X and pixel.Z and a value given by pixel.Value.

"CrystalPeaks[x][z]" has the same definition as in the previous x, z sorting step.

"SizeInX" number of pixels in the x coordinate (128 in this example).

"SizeInZ" number of pixels in the z coordinate (128 in this example).

```
For each pixel in the detector position map do:
begin
    bestDistance = 2 * sizeInX * sizeInX            // set a long best distance
    bestId = 0                                      // set an unused crystal id
    For x = 0 to crystalsInX-1
    begin
      for z = 0 to crystalsInZ-1
      begin
          dx = abs( pixel.X – crystalPeaks[x][z].X)   // delta x
          dz = abs( pixel.Z – crystalPeaks[x][z].Z)   // delta z
          If (dx < sizeInX/3)  and  (dz < sizeInZ/3)  // Is the pixel close to this apex point?
          begin                                       // yes
                distance = dx*dx + dz*dz              // find the distance squared
                If (distance < bestDistance)          // Is this distance less than the current best distance?
                begin                                 // yes
```

```
16                bestDistance = distance              // save as the new best distance
17                bestId = crystalPeaks[x][z].Id       // save the new best crystal id
18              end
19          end
20      end                                            // z loop
21     end                                             // x loop
22    If (bestId > 0)                                  // Was crystal id was found?
23    begin                                            // yes
24      pixel.Value = bestId                           // assign a crystal id to this pixel
25    else
26                                                     //  error – all peaks are too far away
27    end
28  end                                                // pixel loop
```

[0060] Line 11 in the pseudo code above tests to see if the current pixel is close enough to the current peak apex point to go through the expense of a distance check involving multiplication in line 13.

[0061] An example of a detector position map produced by this step is shown in Figure 15. The lines in Figure 15 represent the boundaries between different crystal ID codes. Figure 15 also contains the peak apex pixel locations for reference.

[0062] If the position histogram was compressed initially, the detector position map created in the previous step can be restored to the uncompressed size. In this example, the detector position map coordinates are generated relative to a 128 x 128 size and they can be translated back to a 256 x 256 scale to produce the final detector position map. Pixel replication can be used to uncompress the data. If the peak location is found in floating point, interpolation can be performed at the crystal boundaries. Other methods can also be utilized, such as median filtering.

[0063] Referring back to Figure 2, the detector position map generated according to the above described methods can then be used during a PET scan by the event locator circuit 27 to map the (x, z) coordinate pairs generated by the acquisition circuits 25 to a crystal ID number.

[0064] While the foregoing specification illustrates and describes the preferred embodiments of this invention, it is to be understood that the invention is not limited to the precise construction disclosed herein. The invention can be embodied in other specific forms without departing from the spirit or essential attributes. Accordingly, reference should be made to the following claims, rather than to the foregoing specification, as indicating the scope of the invention.